POLITÉCNICA

"Ingeniamos el futuro"

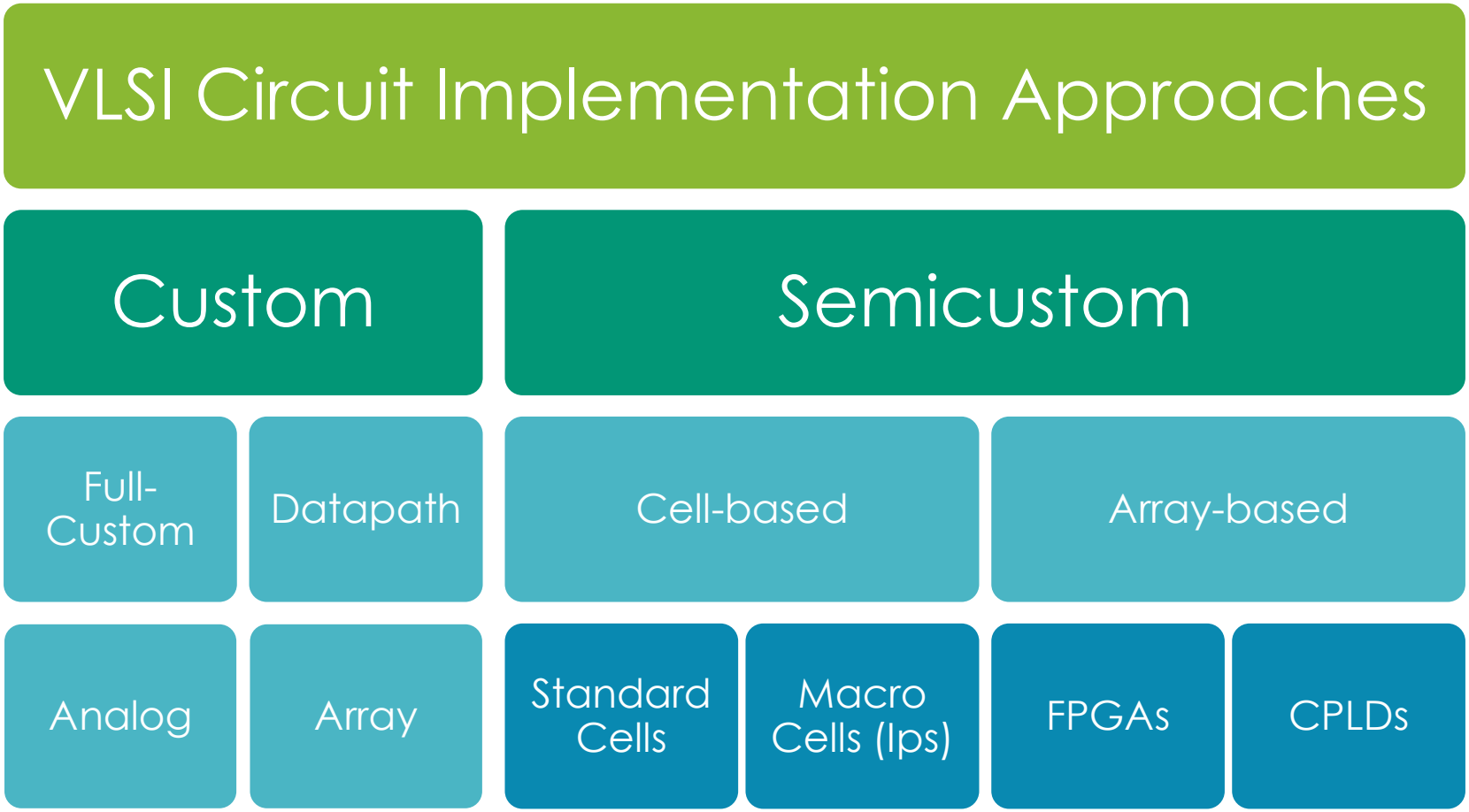
Diseño Físico con Encounter y Scripts

LDIM, Curso 2013-2014

Pablo Ituero

Departamento de Ingeniería Electrónica

VLSI Implementation Choices



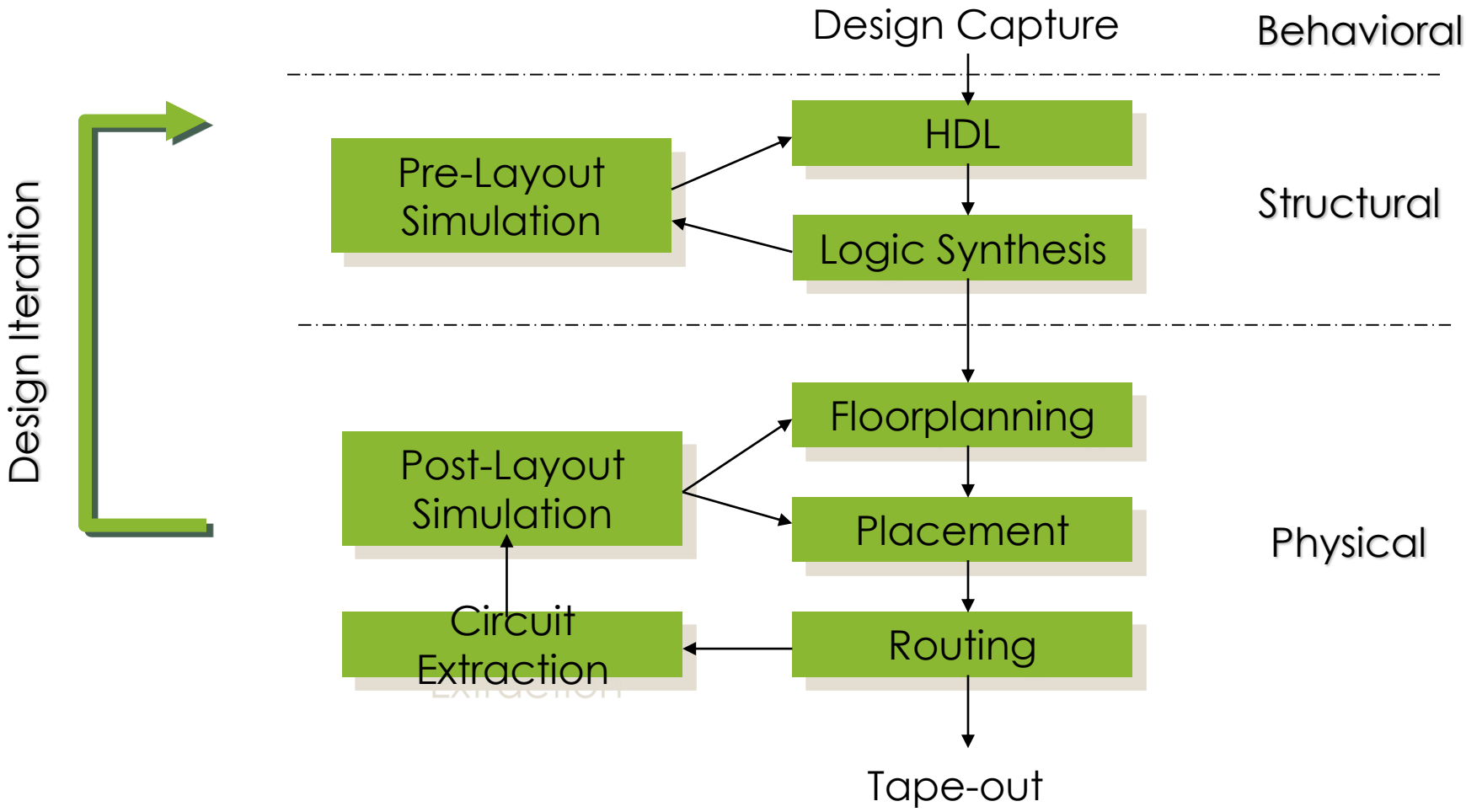
Circuitos Semi-custom

- El esfuerzo de diseño se desplaza a la arquitectura y a la algorítmica (síntesis, colocación, rutado, verificación, …)
- Circuitos *Basados en Células*:
  - Utilizan una biblioteca de células y generadores de memorias.
  - Síntesis desde un código HDL, placement y rutado automático.
  - Actualmente es el tipo de ASIC más popular. Muy buenos resultados de prestaciones en tiempo menor. Prestaciones medias-altas. Tendencia actual para grandes tiradas.
- *FPGAS y CPLD*:
  - Arrays de puertas ya fabricados, toda la parte de interacción con el fabricante, encapsulado, prototipado, testeo, etc., desaparece.
  - Ciclo de trabajo muy rápido. Ideal para prototipado o diseños de tiradas muy bajas.
  - Prestaciones limitadas debido a las estructuras prediseñadas.

Parte 2: Semi-Custom

- Diseño físico digital a partir de un código HDL
- Foco en metodología y automatización
- Desde síntesis hasta place&route.
- Entorno de trabajo profesional. Herramientas CAD de la industria. Librería de células estándar real fabricable AMS 0.35um.
  - Síntesis: Synopsys
  - Diseño físico: Cadence
  - Simulación: Modelsim

Semicustom Design Flow

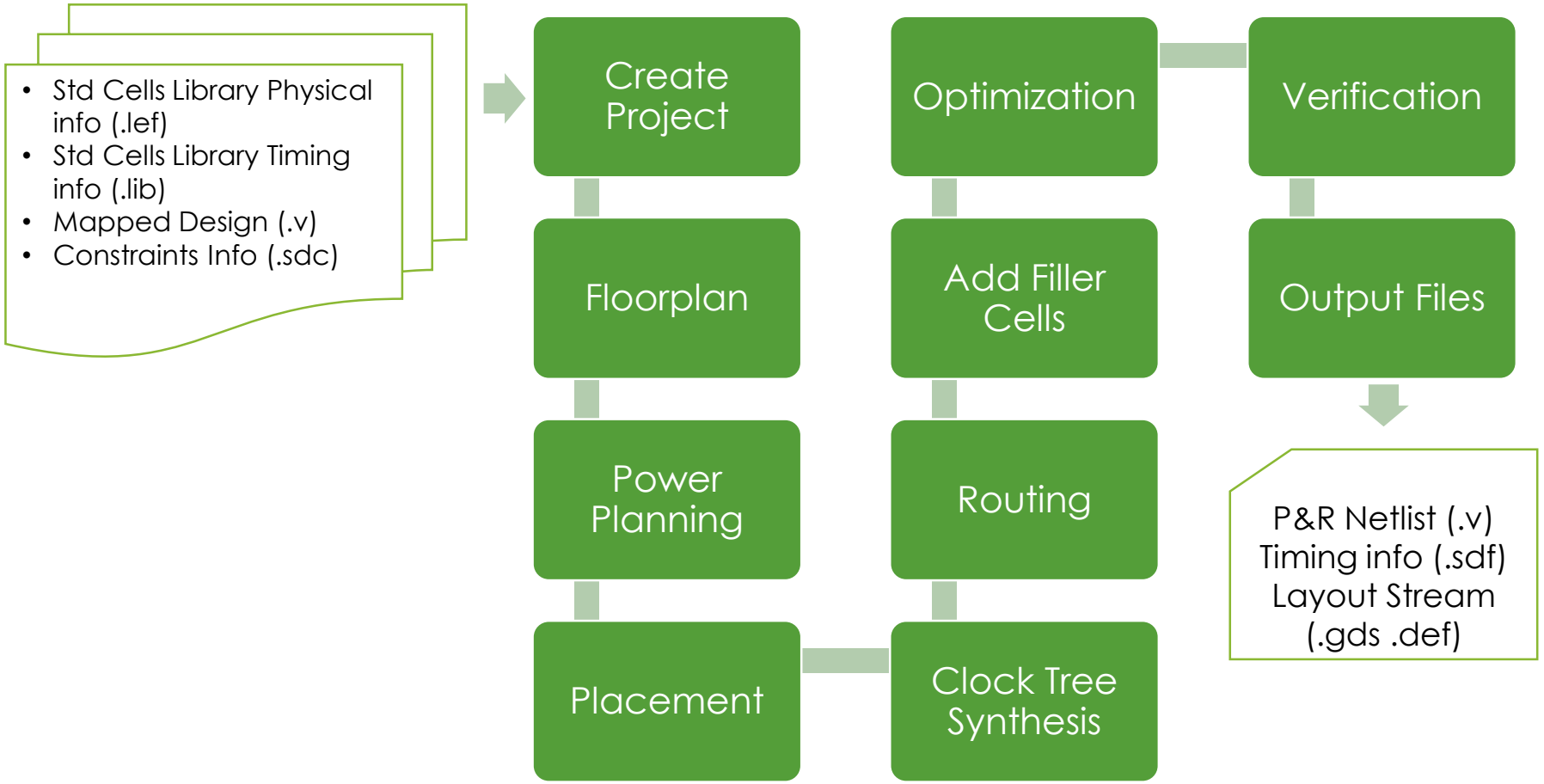


Organización y Evaluación

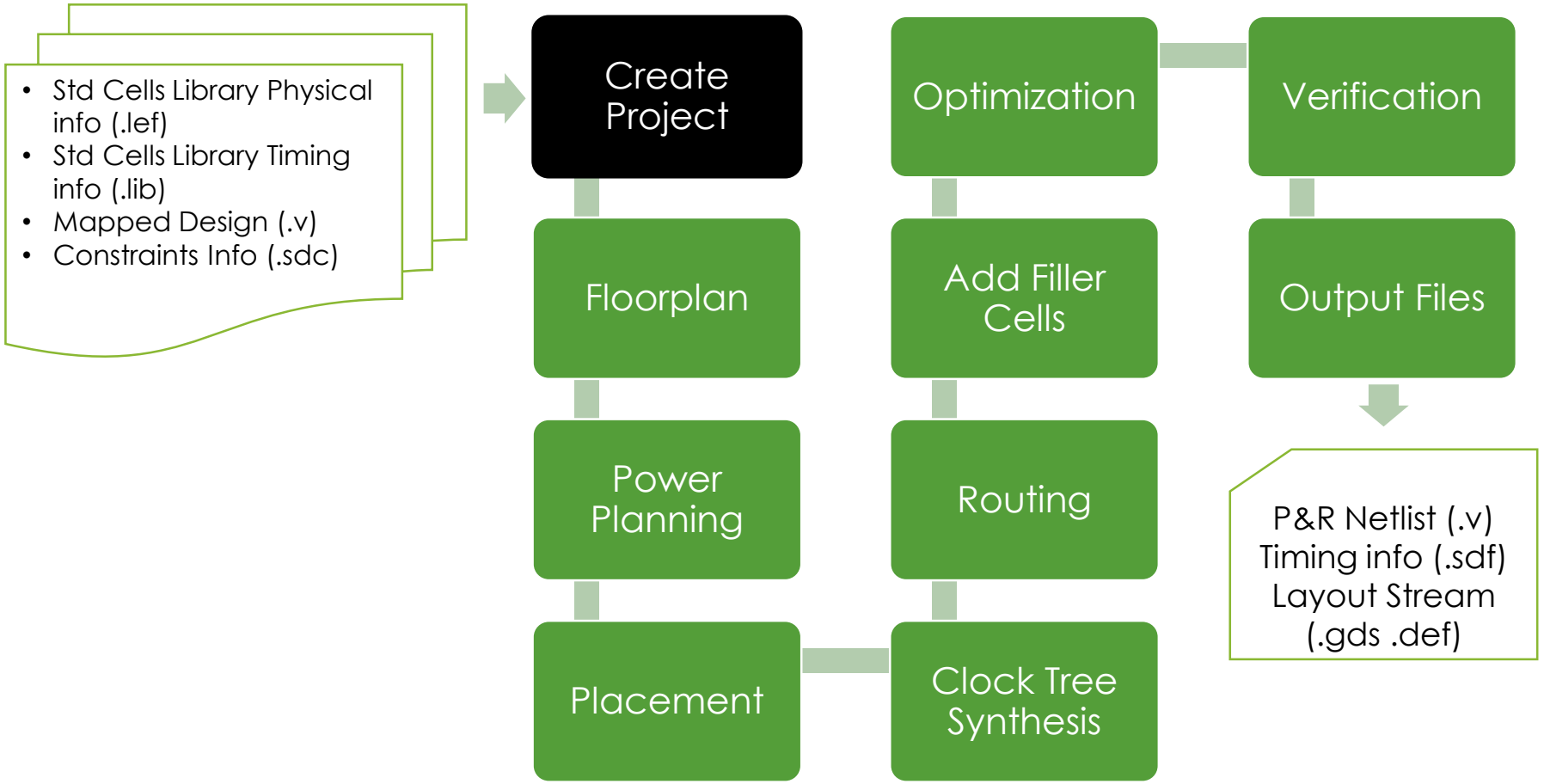
- El laboratorio permanecerá abierto la mañana del jueves de 10 a 13h y todas las tardes de lunes a viernes de 15 a 18h.

- **Parte 1: Diseño Full-Custom**
  - **Semana 0** (10-14 Febrero 2014): Elección de turno e inicio del curso el viernes 15 a las 13.00 en el laboratorio B-043.
  - **Semana 1** (17-21 Febrero 2014): Aprendizaje de la herramienta icfb. Diseño, simulación y caracterización de un inversor. Diseño, simulación y caracterización de dos células básicas: NAND, NOR de dos entradas o similar.
  - **Semana 2** (24-28 Febrero 2014): Parámetros, análisis de corners y análisis estadístico.
  - **Semana 3** (3-7 Marzo 2014): Trazados, DRC, LVS y backannotation.
  - **Semanas 4-6** (10-28 Marzo 2014) Realización de la práctica final de la parte 1, diseño, simulación, caracterización y trazado de un bloque de complejidad media-baja.
- **Parte 2: Diseño Semi-Custom**
  - **Semana 7** (31 Marzo - 4 Abril 2014): Síntesis lógica con Synopsys. Simulación. Optimización de consumo.
  - **Semana 8** (6-11 Abril 2014): Colocación y rutado con Encounter. Utilización de scripts.
  - **Semanas 9-10** (22 Abril - 6 Mayo 2014): Realización de la práctica final de la parte 2. Diseño físico de un circuito digital de complejidad alta descrito en VHDL.

Physical Design Flow



Physical Design Flow



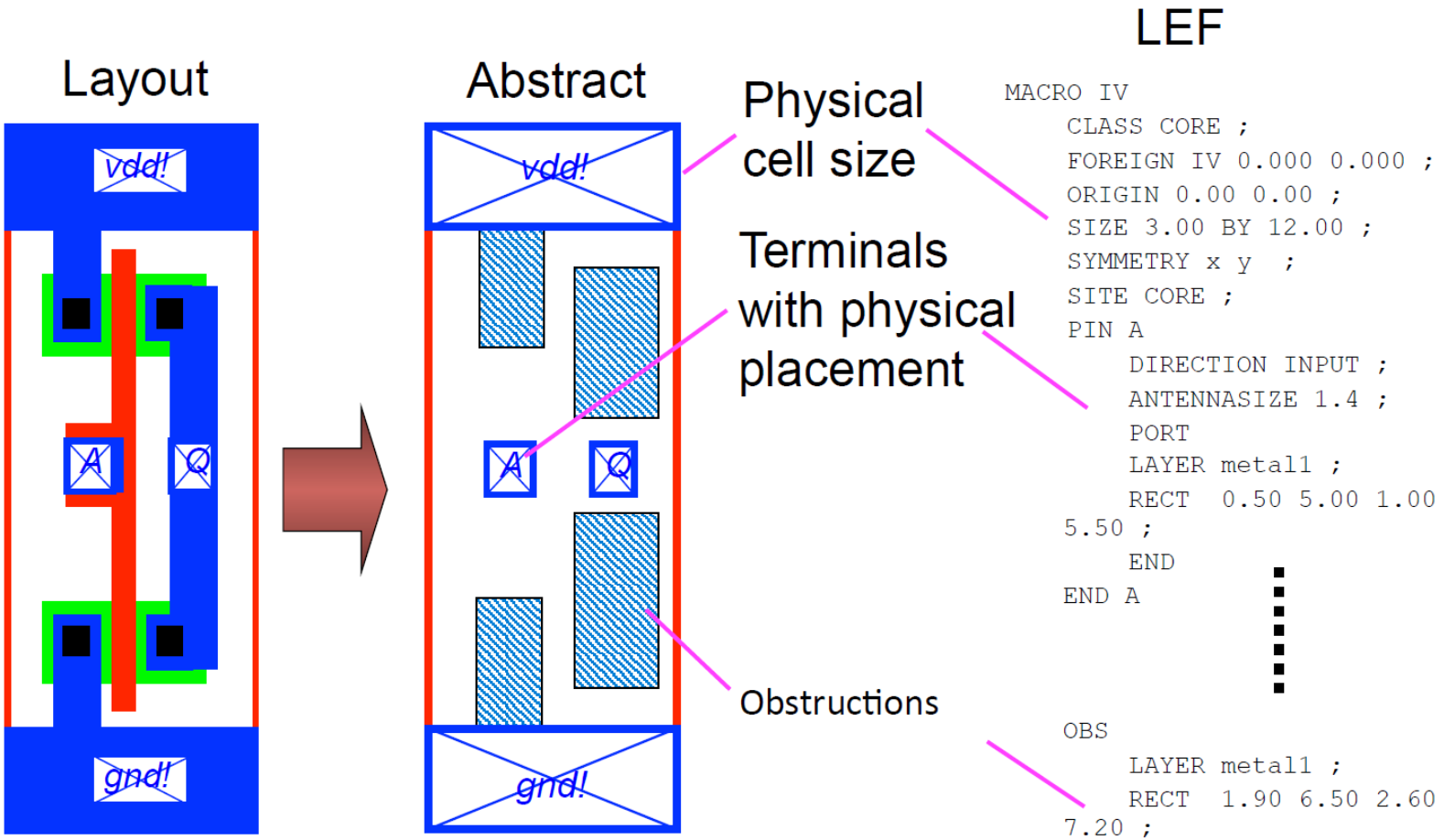
Required Data for P&R

- LEF: Library Exchange Format
  - Cell Library
  - Pad Library
  - IP Cores
- LIB: Liberty File with timing information
- SDC: Synopsys Design Constraints
- V: Mapped design

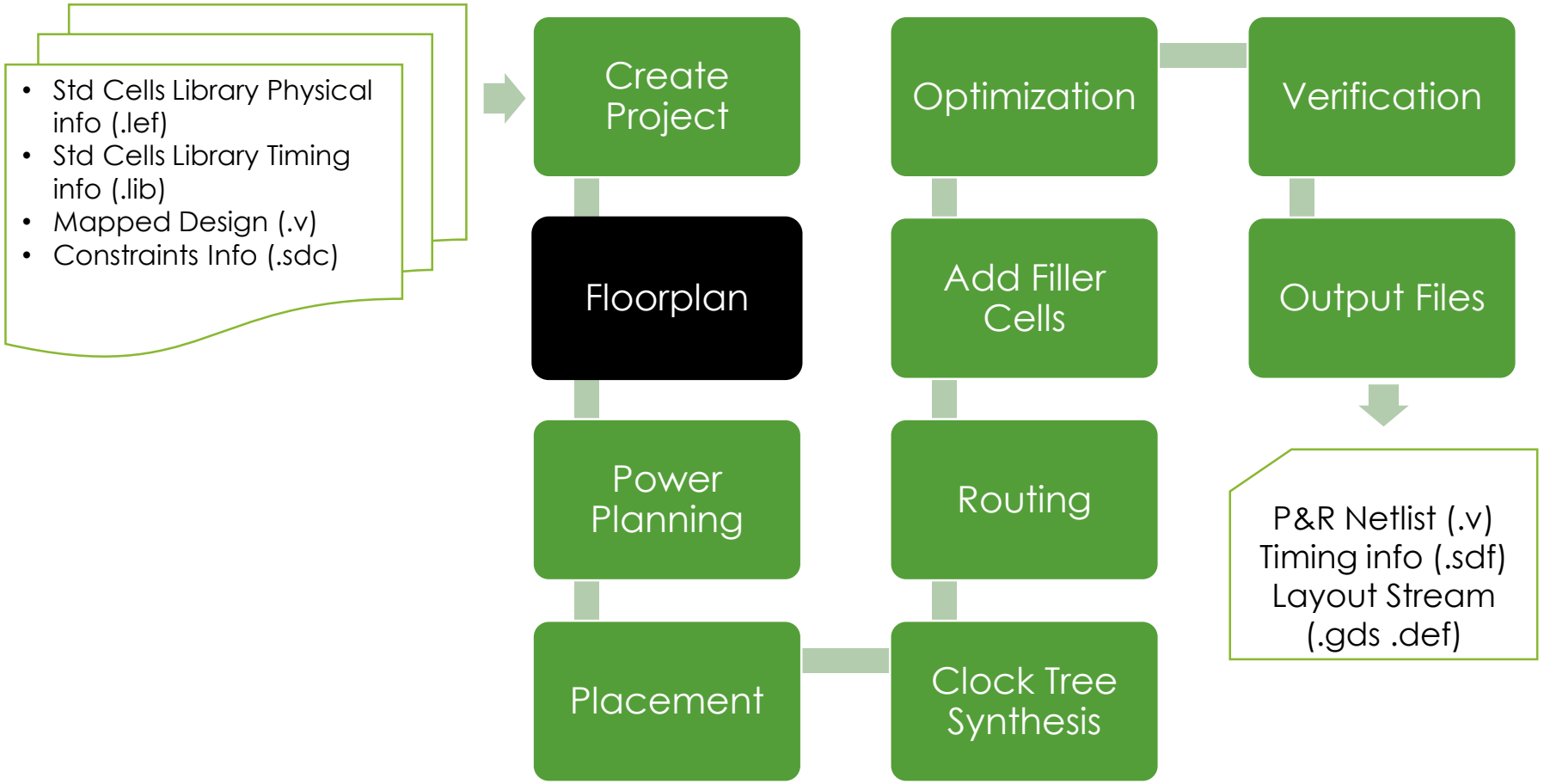
Technology Description Files

- LEF: Library Exchange Format
  - Technology: Design rules, capacitance, resistance, antenna factors, vias.
  - Cells and pads: Size, class, placement info, pin info, obstructuions.

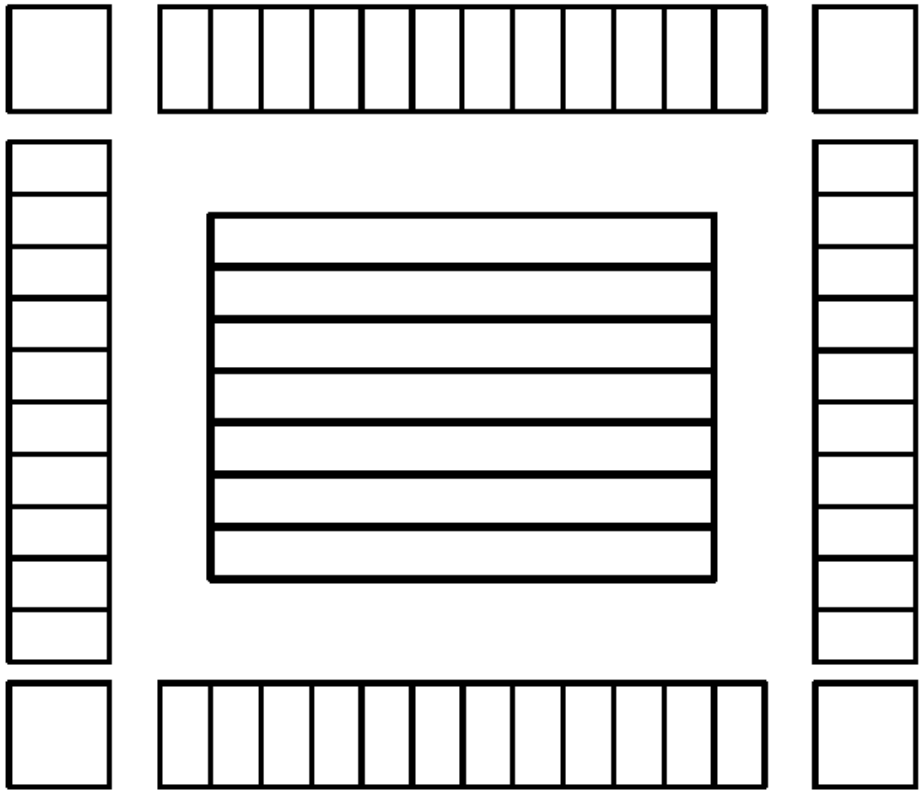
LEF-Example: Inverter



Physical Design Flow

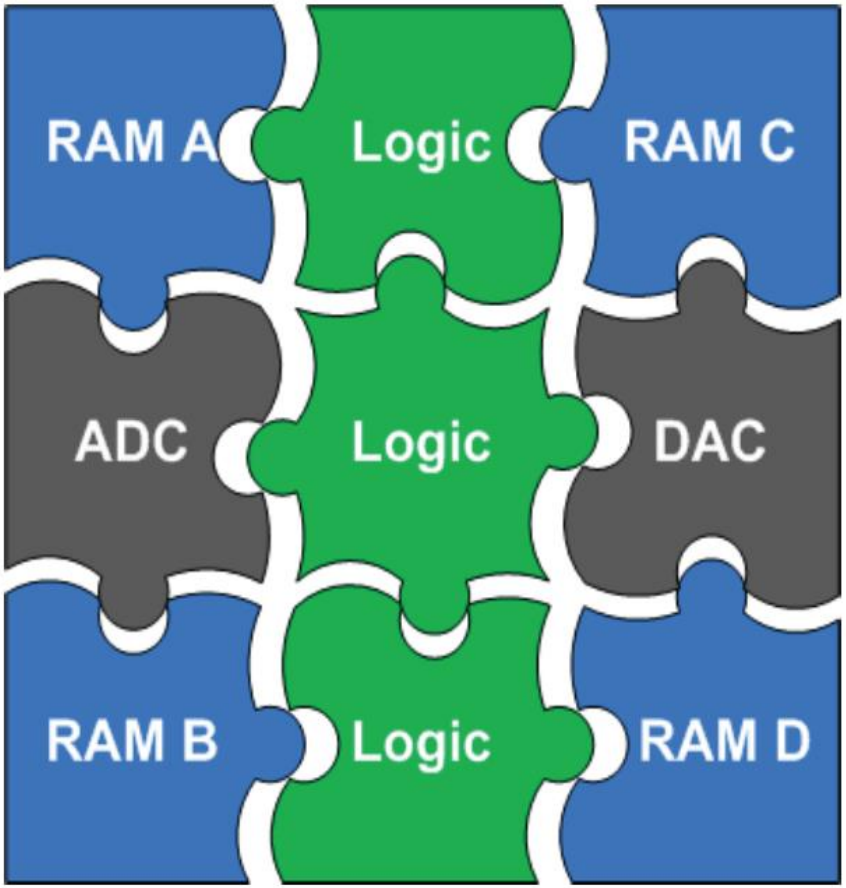


Floorplan



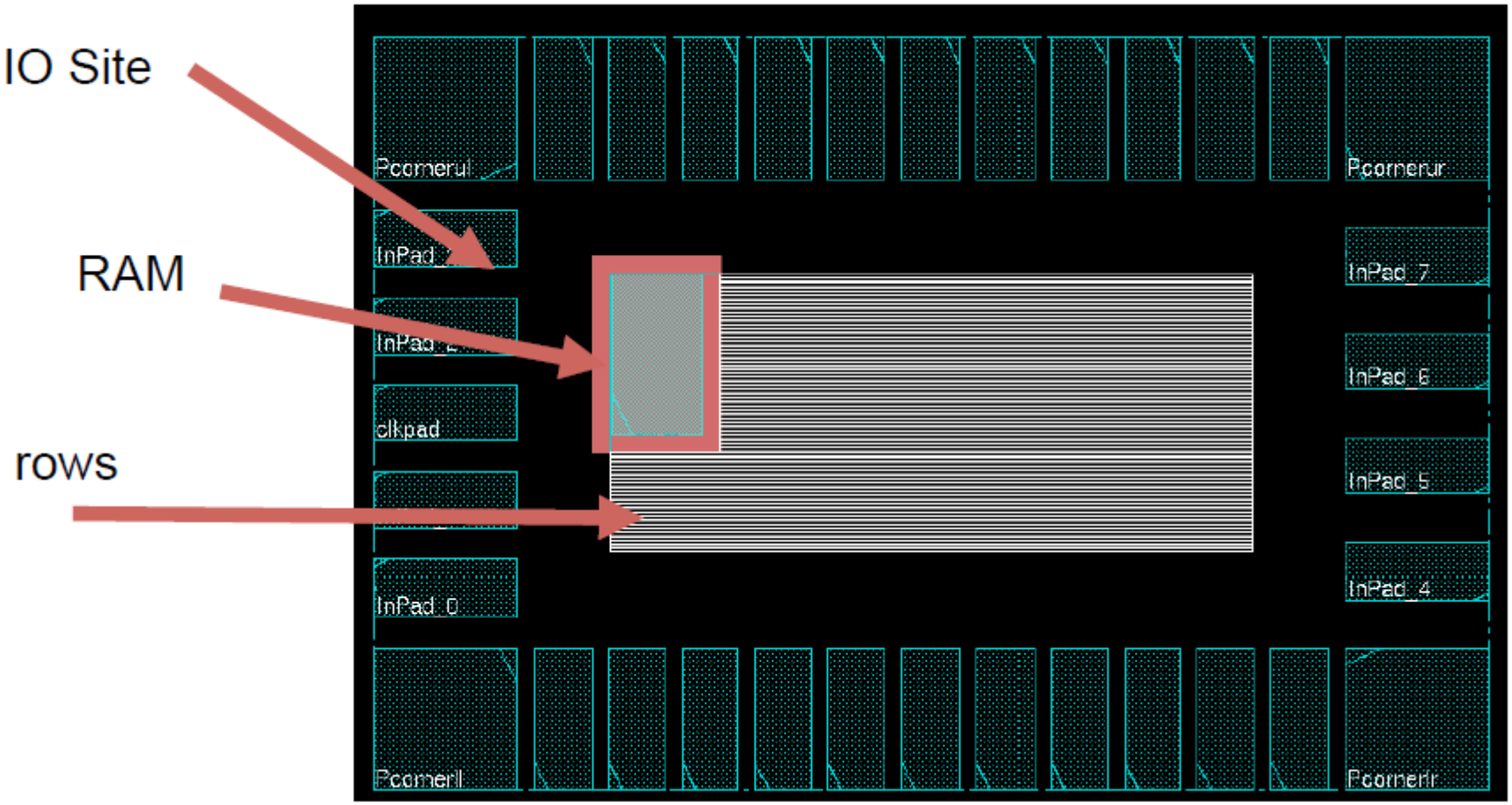
- Floorplan:
  - Placement area
  - IOs
  - RAM/ROM

Floorplan

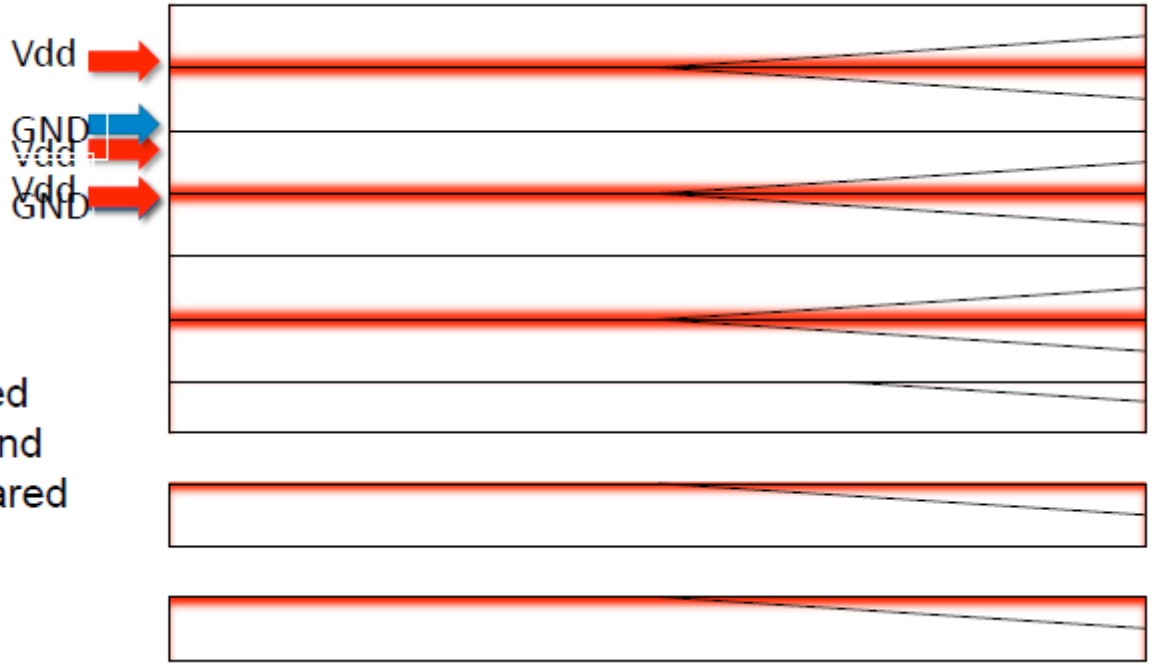


- A starting floorplan is created (required area is estimated by the tool)
- Global de detailed routing grids are created
- The core rows are created
- Sites for IOs are created
  - IO and block to core distance is defined by the user

Floorplan

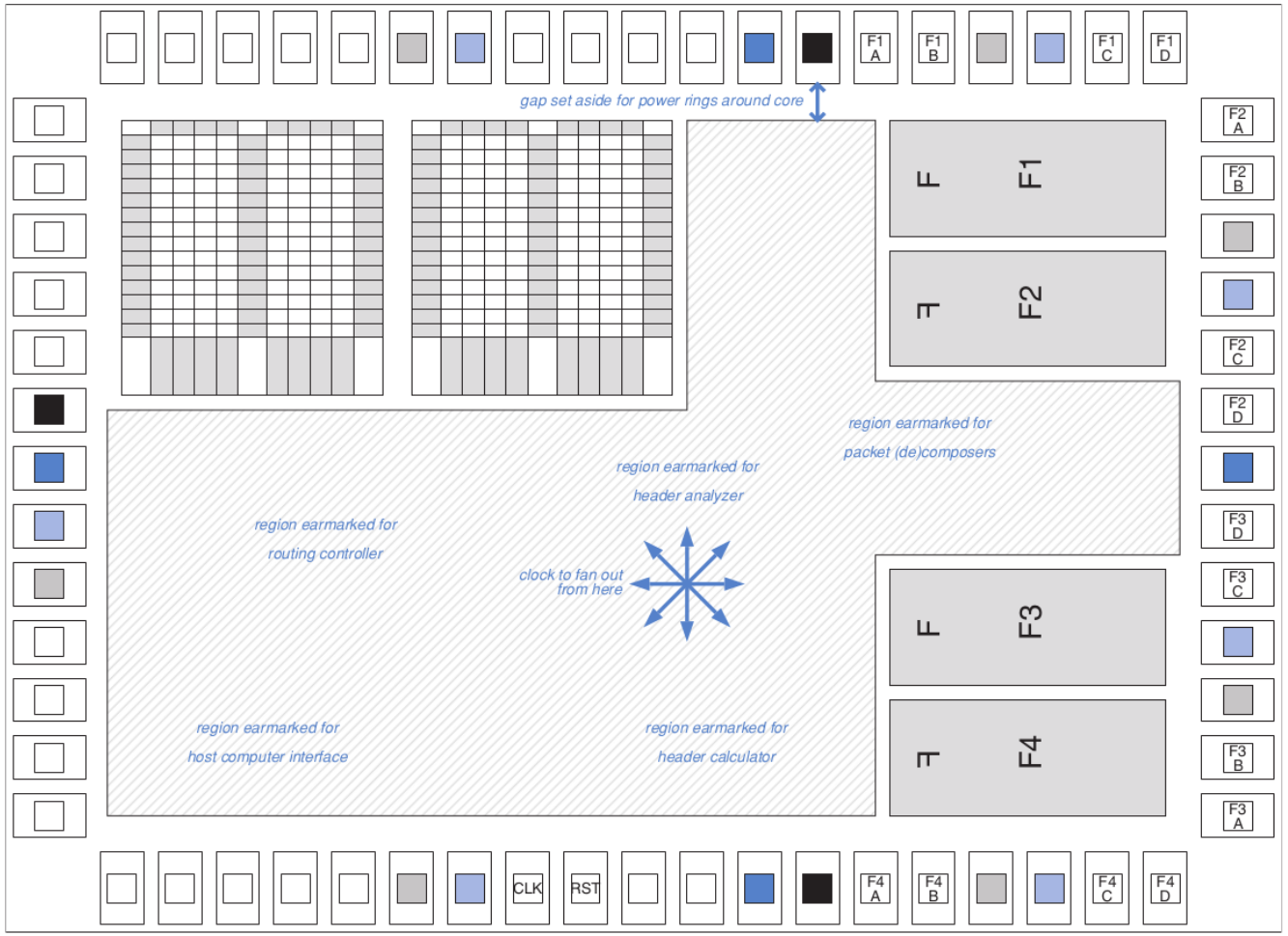

IO Site
RAM
rows
Pcornerul
Pcornerur
InPad_7
InPad_6
InPad_5
InPad_4
clkpad
InPad_0
Pcornerll
Pcornerlr

Cell Rows

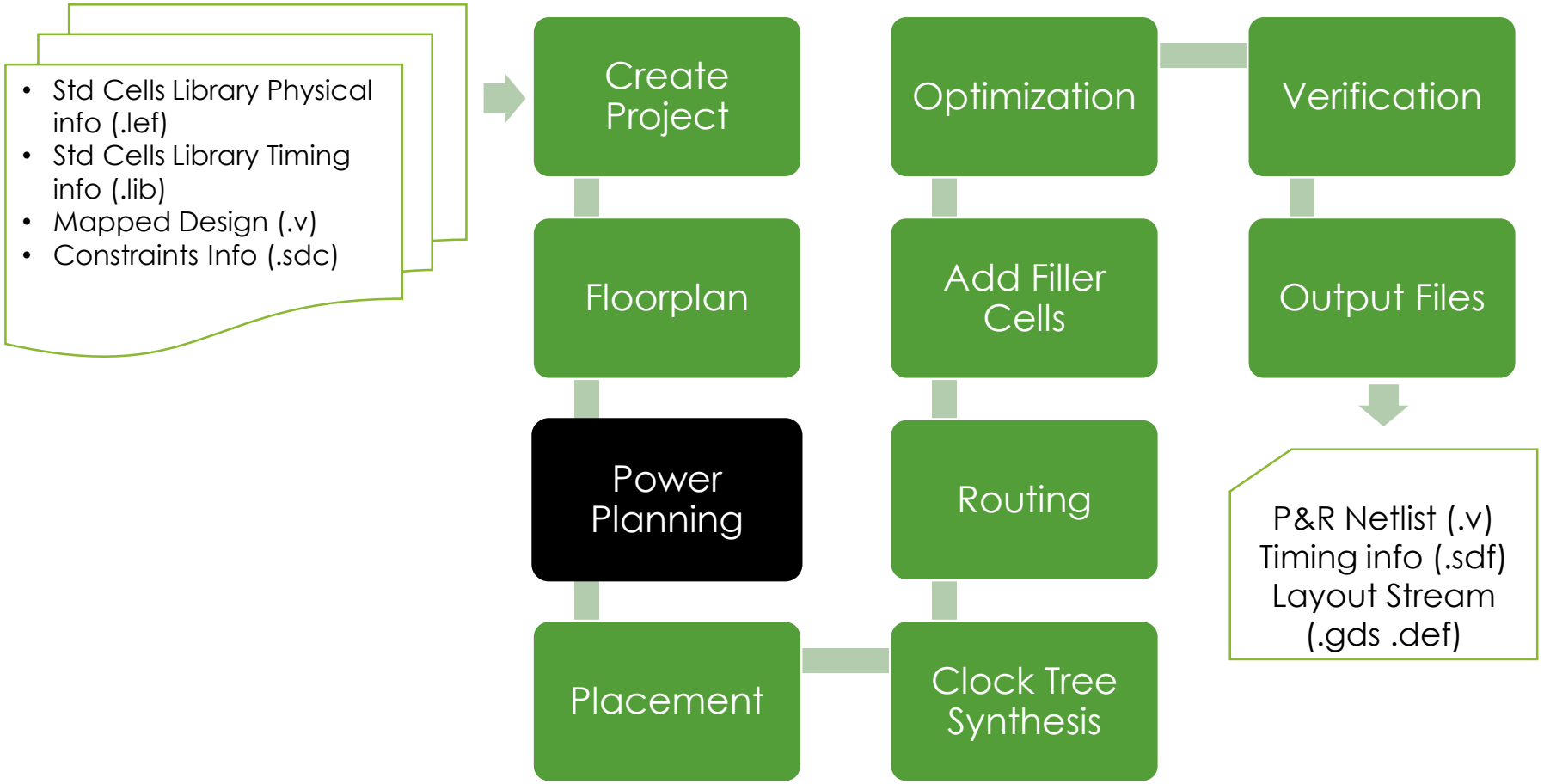


If rows are flipped and abut VDD and GND can be shared by 2 rows.
Default setting!

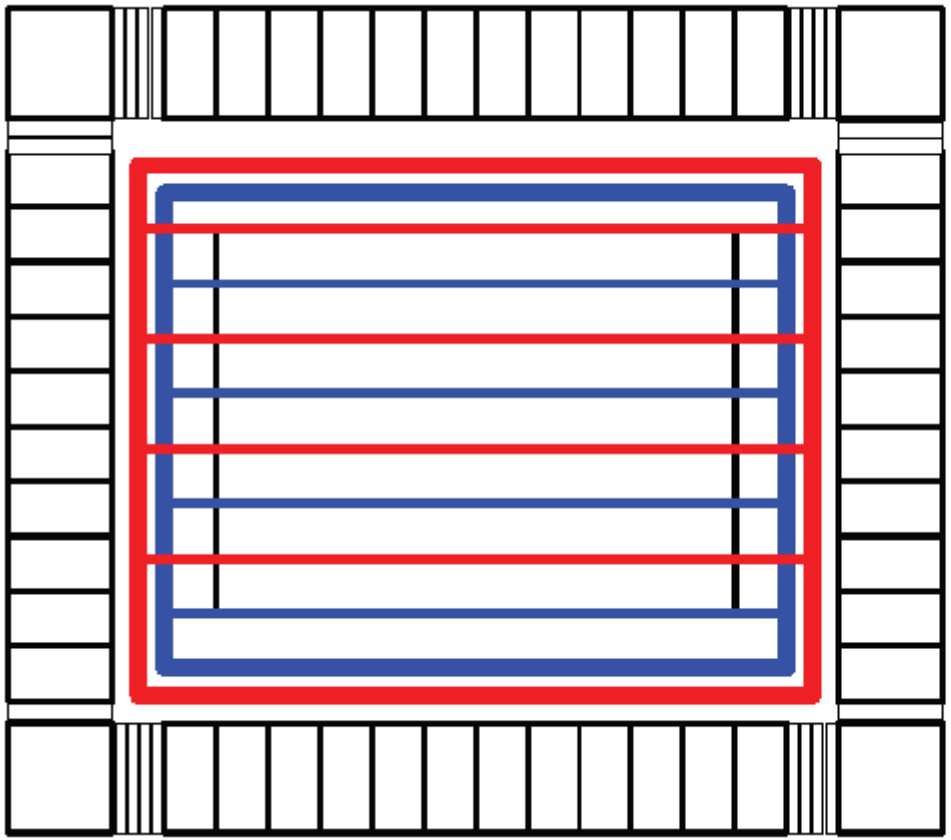
Floorplan


F1 A
F1 B
F1 C
F1 D
gap set aside for power rings around core
F2 A
F2 B
F2 C
F2 D
F F1
F F2
region earmarked for
packet (de)composers
region earmarked for
header analyzer
region earmarked for
routing controller
clock to fan out
from here
F3 D
F3 C
F F3
F F4
region earmarked for
host computer interface
region earmarked for
header calculator
F3 B
F3 A
CLK
RST
F4 A
F4 B
F4 C
F4 D
core supply e.g. 1.8V
core ground 0V
padframe ground 0V
padframe supply e.g. 3.3V

Physical Design Flow



Power Planning

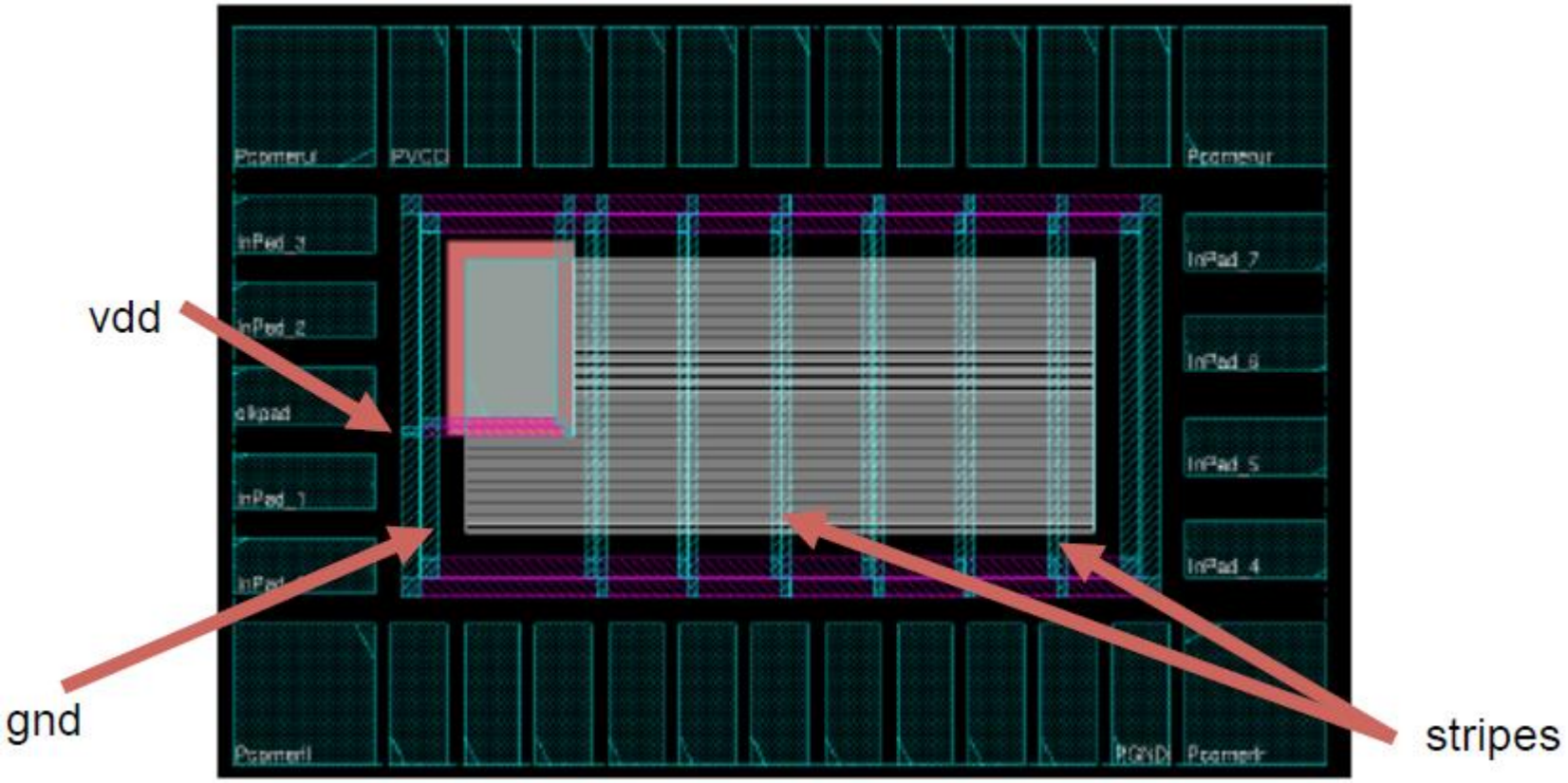


- Power Planning
  - Design a power ring
  - Add horizantal and vertical power stripes

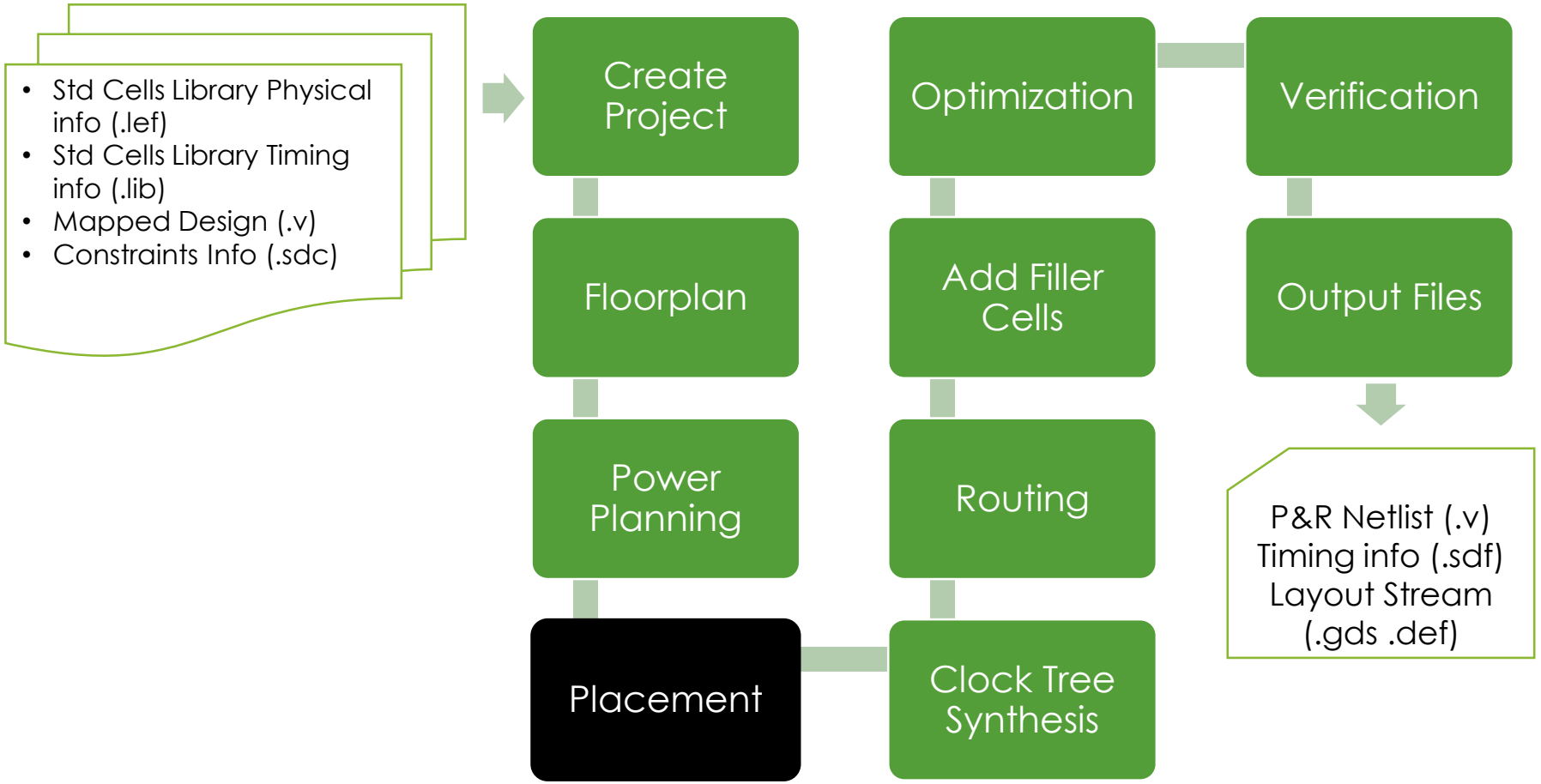
Power Planning

- Power paths are planned and modified before routing
- Creation of power rings that surround all block and core
- Creation of stripes over rows
- Connects rings, stripes and pads

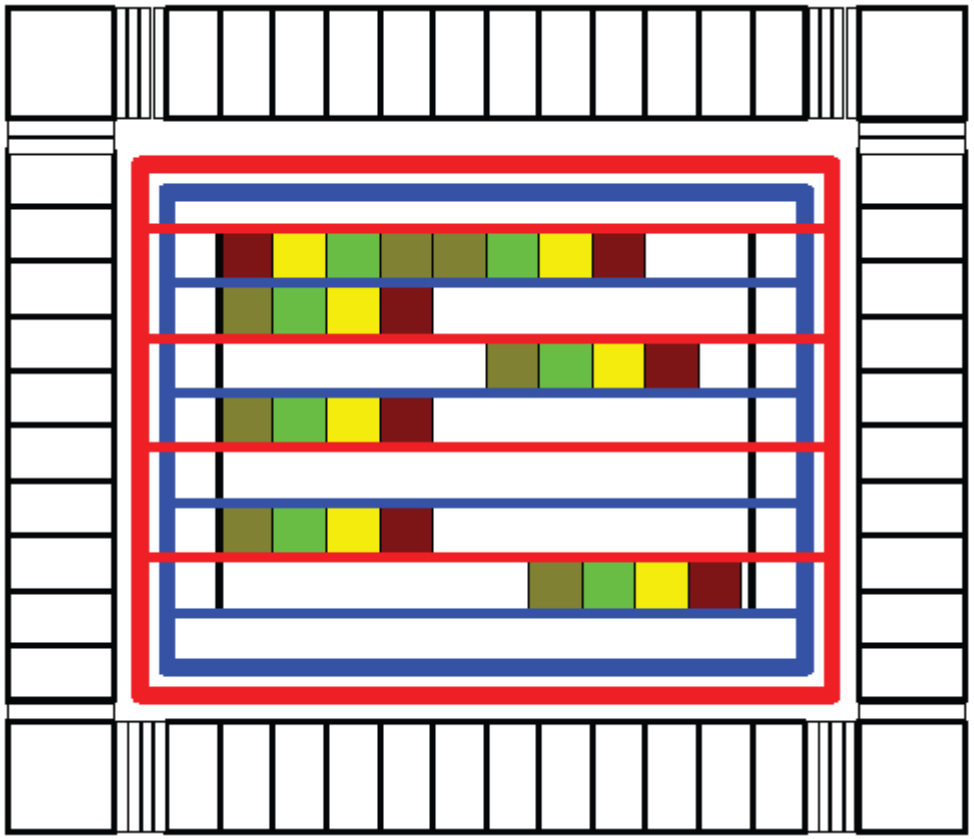
Power Planning



Physical Design Flow



Placement

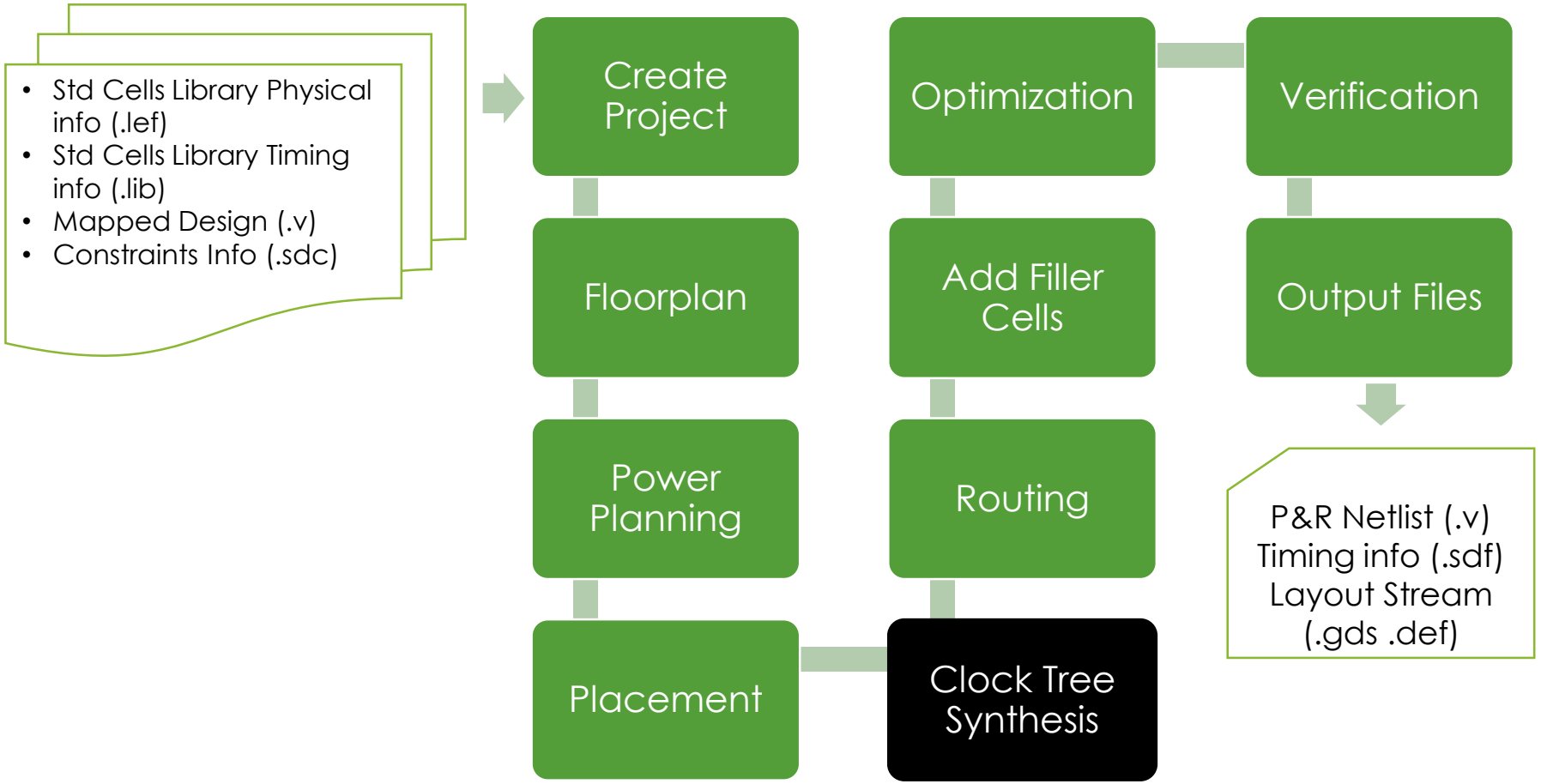


- Place Cells:
  - Place all the standard cells into the rows

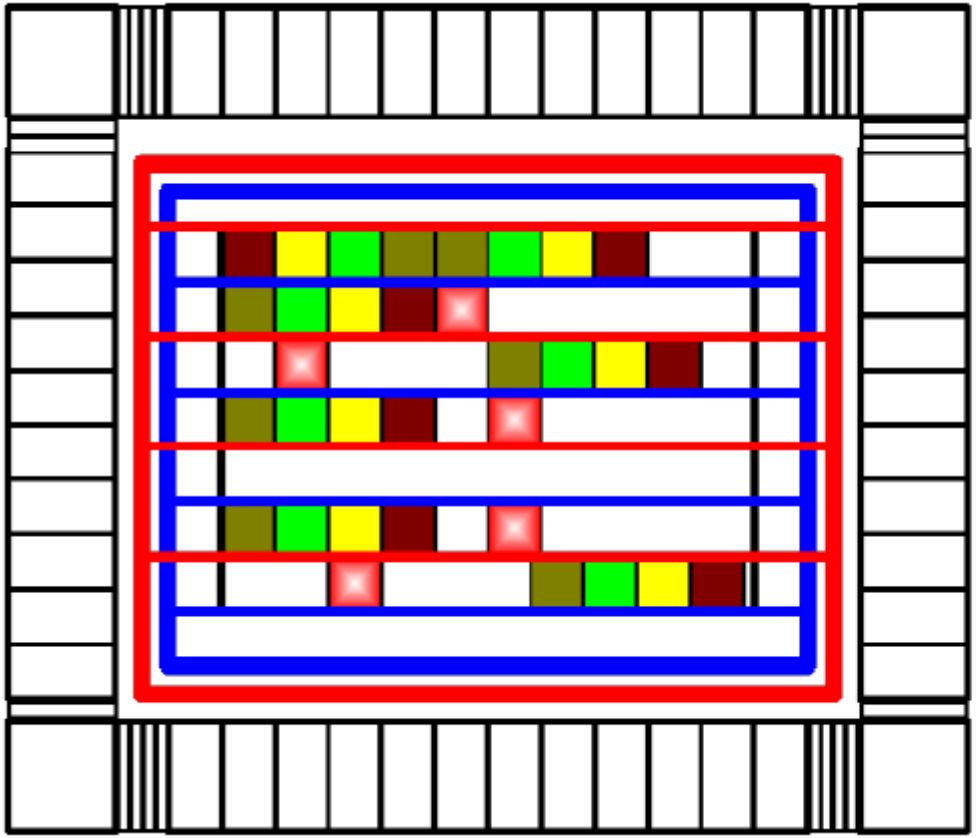
Placement

- Initial cell placement
- Moves, swaps, changes of orientation of cells to minimize required wire length
- Optimize for wire length and net crossings

Physical Design Flow



Clock Tree Synthesis

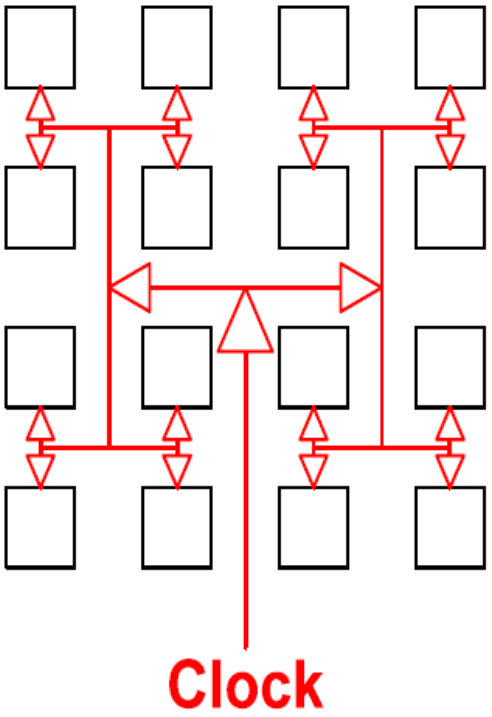


Clock Tree Synthesis:

- Places clock buffers
- Timing constraints
  - Skew etc

Clock Tree Synthesis

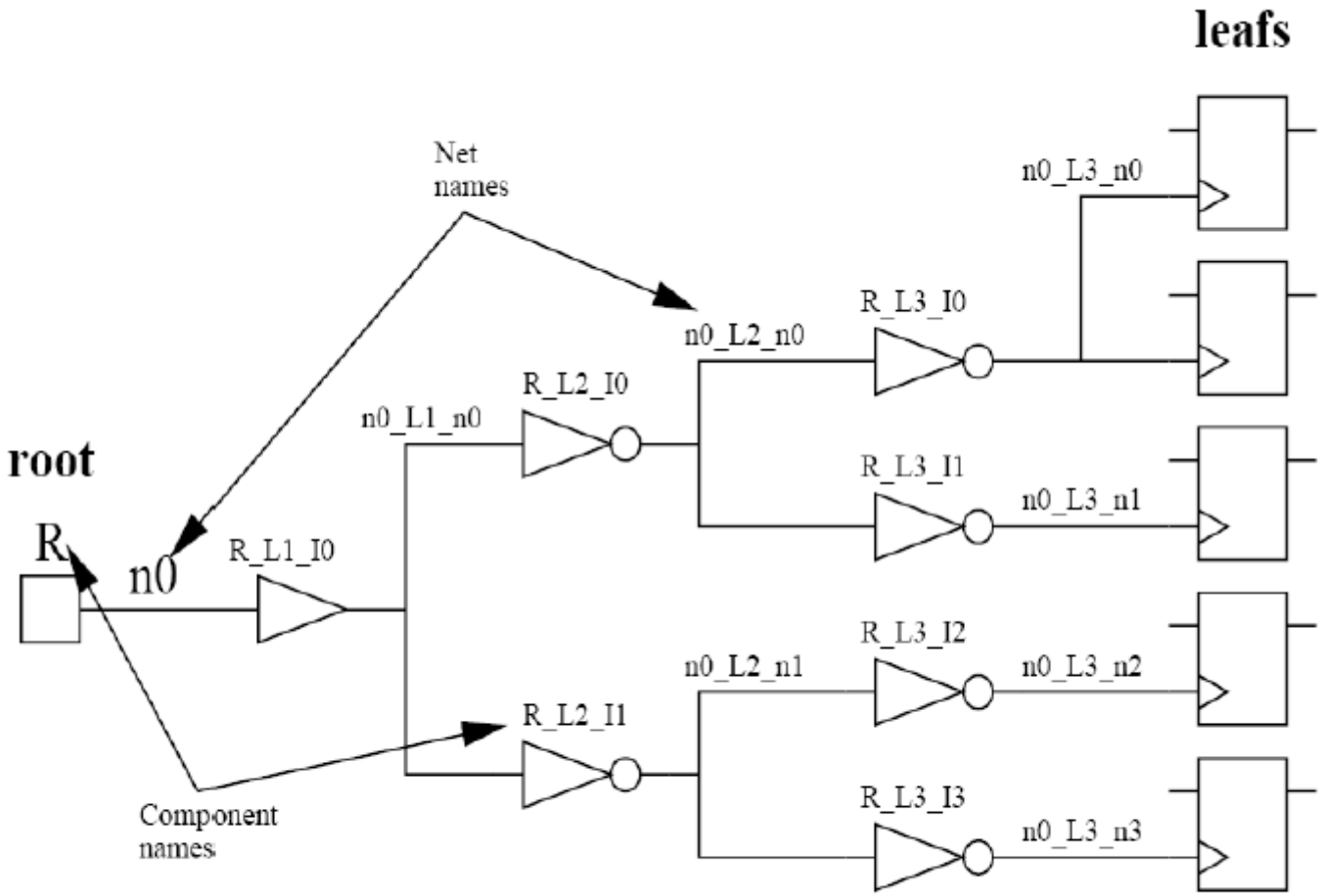
- Clock tree is synthesized and routed with highest priority to minimize clock skew



Small relative skew

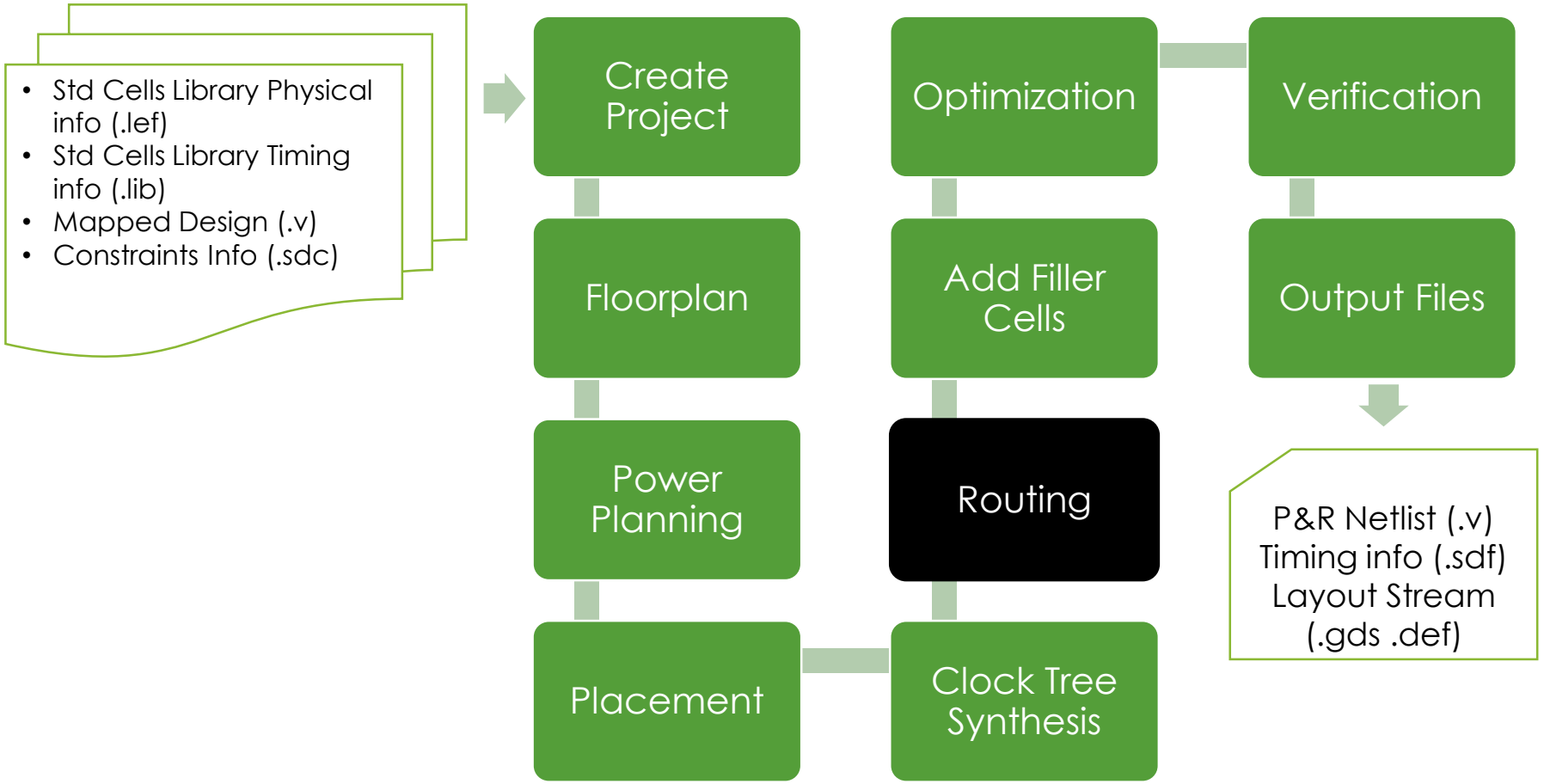
Absolute skew of less importance

Clock Tree Synthesis



Clock buffers are placed in the core row gaps

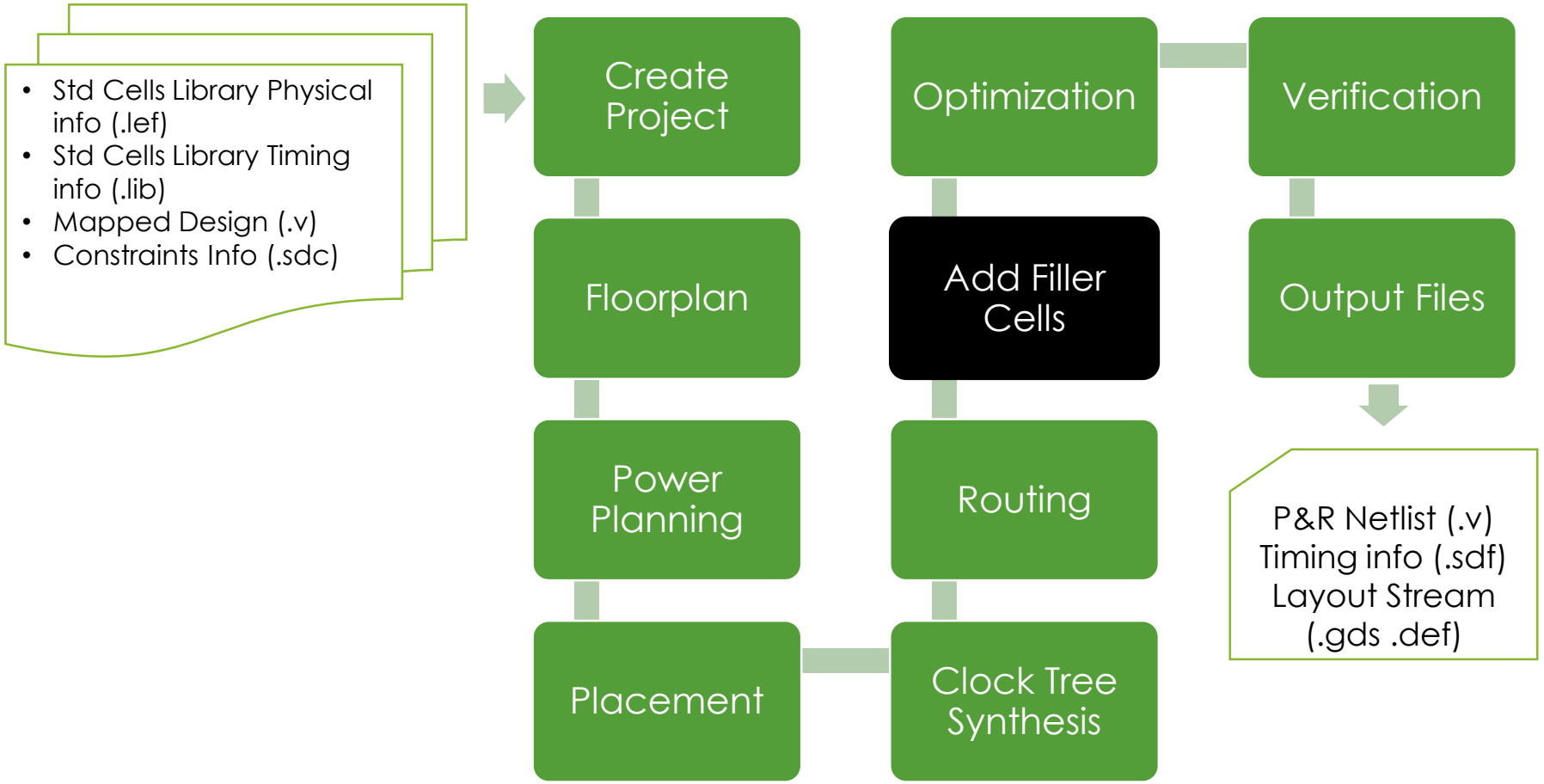
Physical Design Flow



Routing

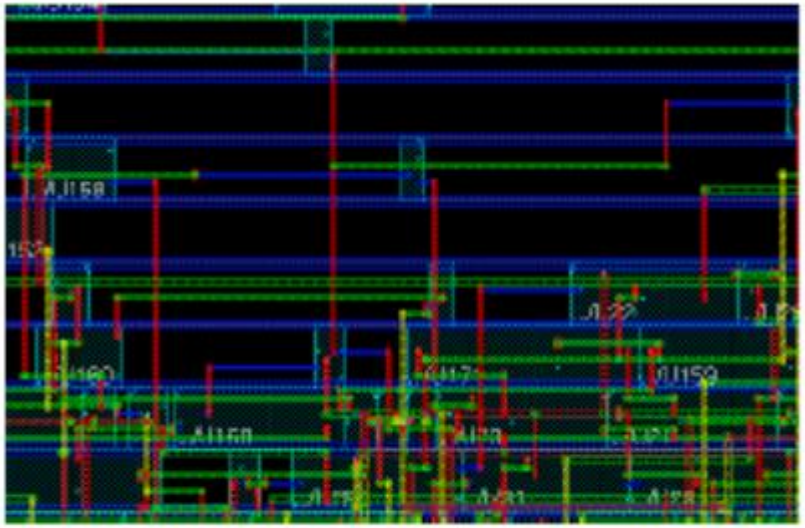
- Signal routing
  - Connects cells according to netlist
  - Metal wires are connected over several layers
- Routing time is strongly dependent on the design complexity

Physical Design Flow

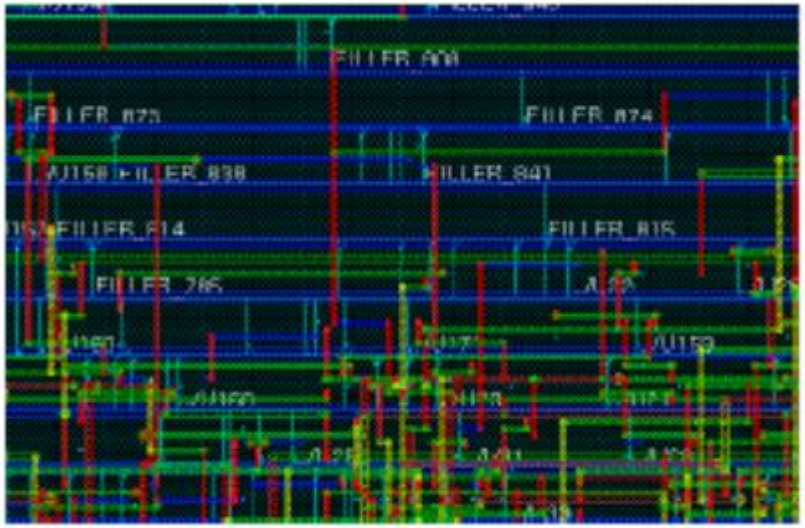


Add Filler Cells

- Core filler cells ensure the continuity of power/ground rails and N/P wells in the row

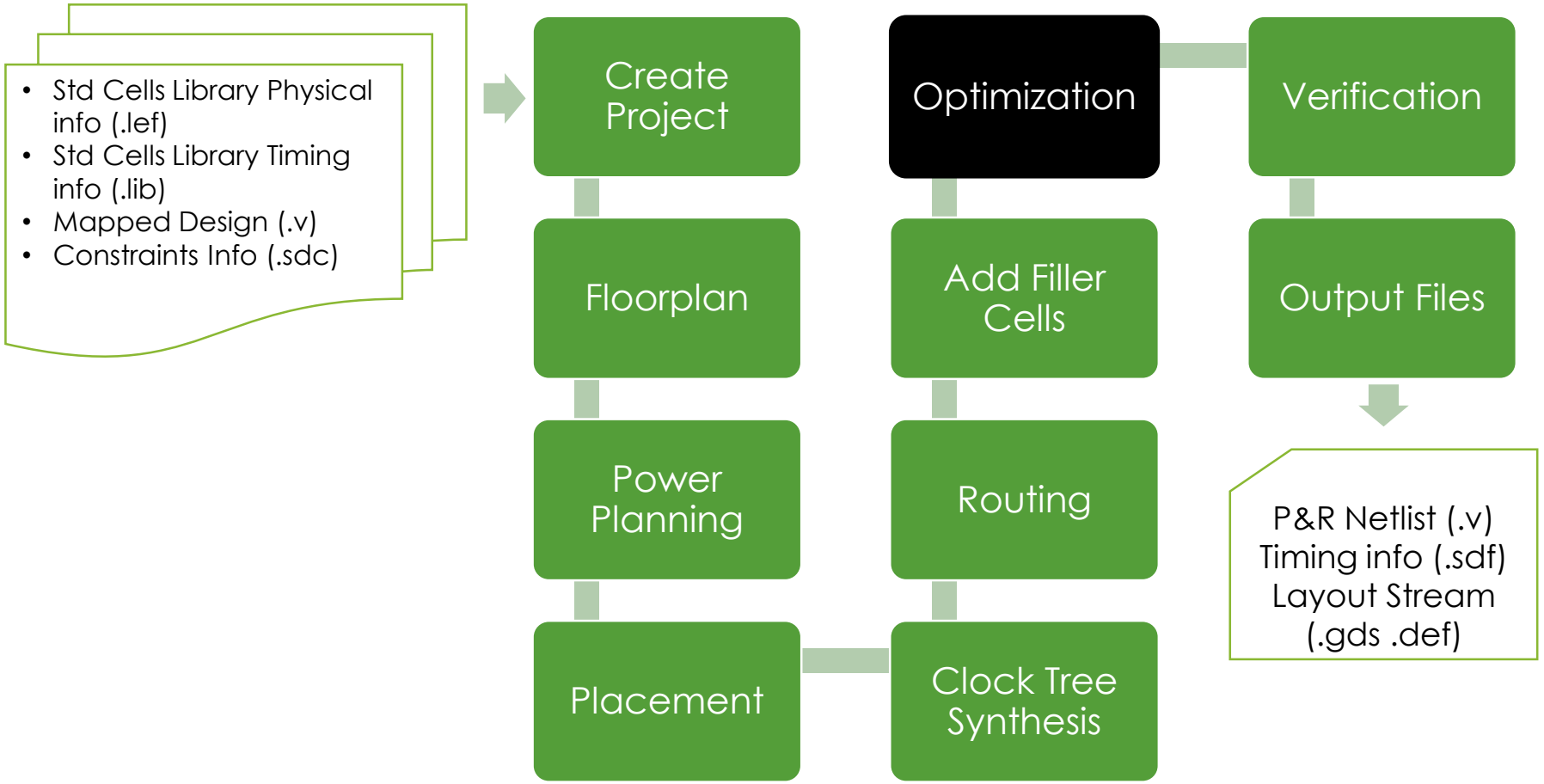


before



after

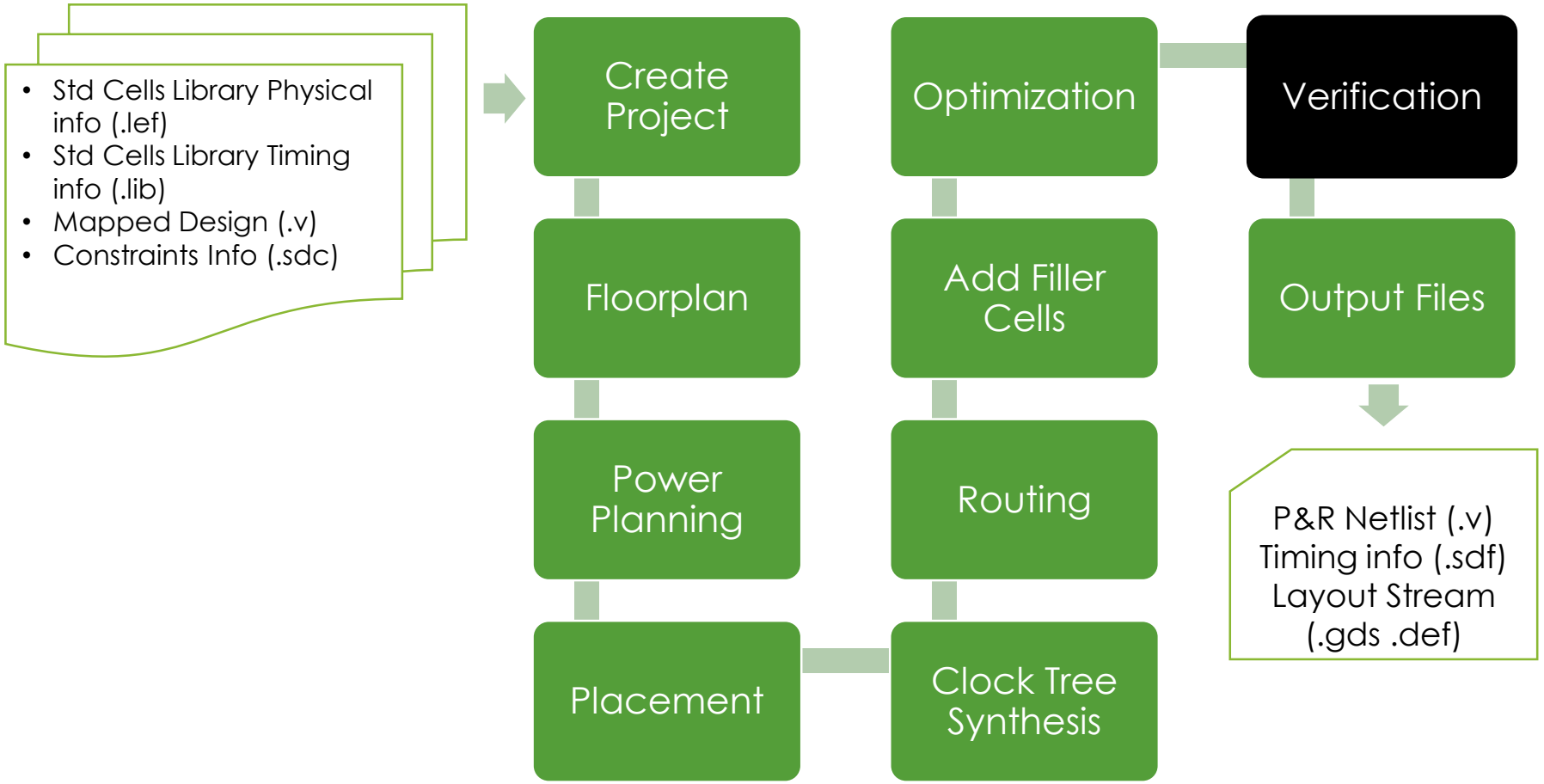
Physical Design Flow



Optimization

- Timing optimization
- Optimize the design based on actual wires.
- Any modified net will be automatically re-routed

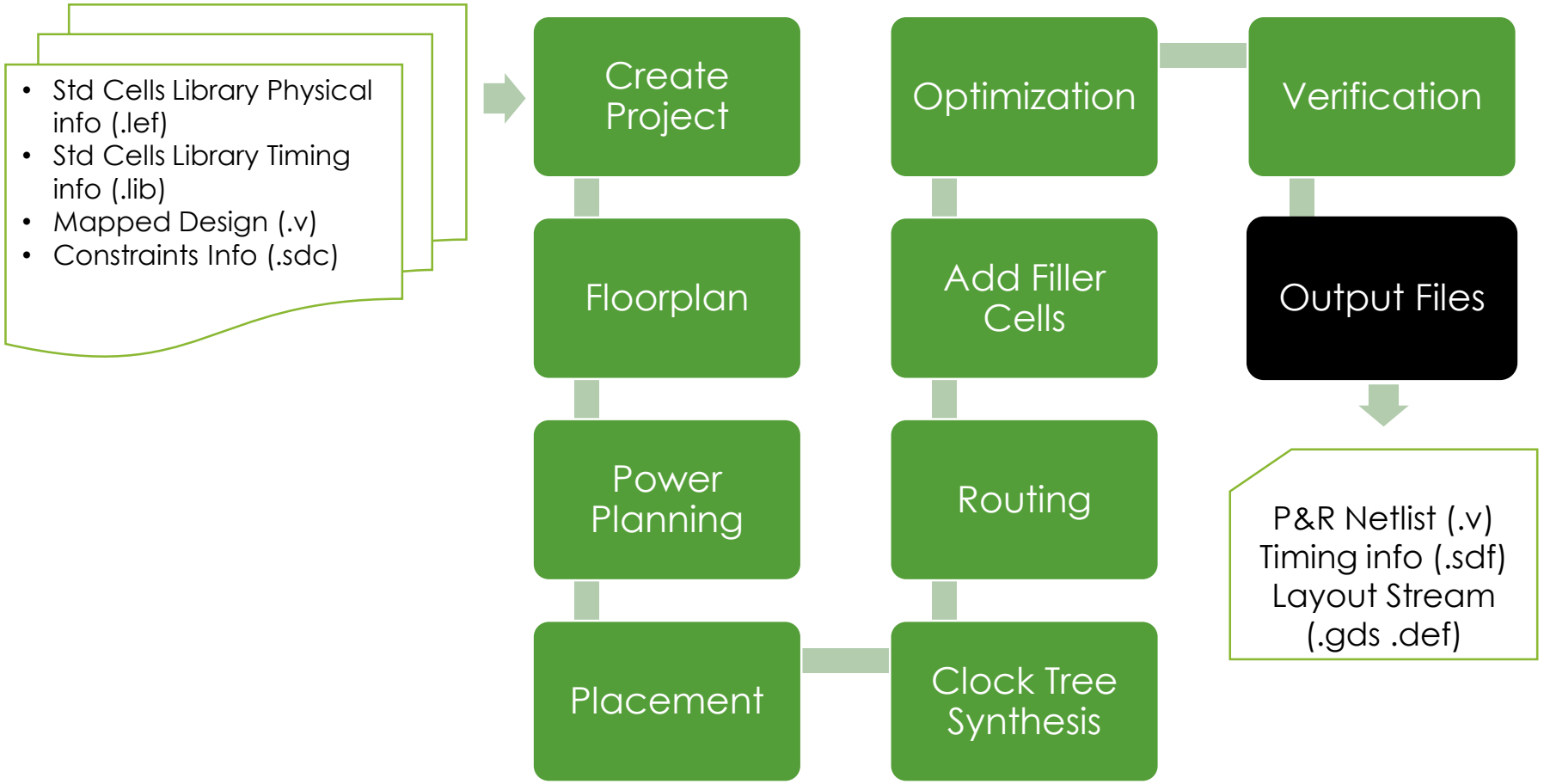
Physical Design Flow



Verification

- Connectivity
- DRC
- Antenna

Physical Design Flow



Output Files

- .v: P&R netlist
- .sdf: Timing information
- .gds or .def: Layout stream

Scripts

- Each action in the GUI corresponds to a certain command that can be written in a script file.
- Programs keep a log of the commands in each session.
- Advantages of using scripts:
  - Easy to change parameters
  - Improved reusability
  - Reduced time
  - Serve as documentation

.synopsys_dc.setup

```
set search_path [list . ${synopsys_root}/libraries/syn
${synopsys_root}/minpower/syn ${synopsys_root}/dw/syn_ver
${synopsys_root}/dw/sim_ver /opt/ams/ams3.8/synopsys/c35_3.3V]

set target_library  [list
/opt/ams/ams3.8/synopsys/c35_3.3V/c35_CORELIB_TYP.db]

set synthetic_library [list dw_foundation.sldb]

set command_log_file  "./command.log"

set designer  ""

set company  "Universidad Politecnica de Madrid"

set find_converts_name_lists  "false"

set symbol_library  [list generic.sdb]

set link_library  [ concat [concat "*" $target_library]
$synthetic_library]
```

syndc.tcl

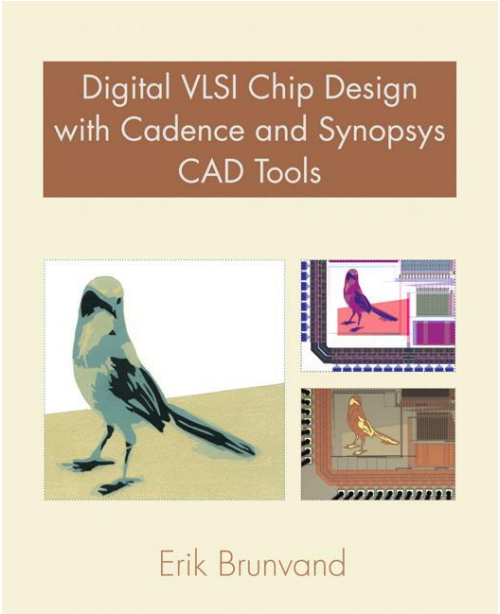
```
#################################################################

# below are parameters that you will want to set for each design

#################################################################

set myFiles ~/Semi-Files/CONTADOR.vhdl ;# list of all HDL files in the design

set fileFormat VHDL              ;# verilog or VHDL

set basename CONTADOR           ;# Top-level module name

set runname v1                 ;# Name appended to output files

set myClk clk                     ;# The name of your clock

set virtual 0                       ;# 1 if virtual clock, 0 if real clock

set parameters 1                    ;# 1 if there are parameters to set, 0 if not

set myParameters "WIDTH = 12"       ;# list of your parameters


# Timing and loading information

set myPeriod_ns 10               ;# desired clock period (in ns) (sets speed goal)

set myInDelay_ns 0              ;# delay from clock to inputs valid

set myOutDelay_ns 0             ;# delay from clock to output valid

set myInputBuf INV4            ;# name of cell driving the inputs

set myLoadLibrary  c35_CORELIB_TYP      ;# name of library the cell comes from

set myLoadPin A             ;# name of pin that the outputs drive

# the following control which output files you want. They

# should be set to 1 if you want the file, 0 if not


set write_v 1            ;# compiled structural Verilog file

set write_ddc 1          ;# compiled file in ddc format (XG-mode)

set write_sdf 1          ;# sdf file for back-annotated timing sim

set write_sdc 1          ;# sdc constraint file for place and route

set write_rep 1          ;# report file from compilation

set write_pow 0          ;# report file for power estimate


# compiler switches...

set useUltra 1                      ;# 1 for compile_ultra, 0 for compile

                                     # mapEffort, useUngroup are for

                                     # non-ultra compile...

set mapEffort1 medium               ;# First pass - low, medium, or high

set mapEffort2 medium               ;# second pass - low, medium, or high

set useUngroup 1                    ;# 0 if no flatten, 1 if flatten
```

top-level.tcl

```
# set the BASENAME for the .v and .sdc files generated by
Synopsys.

# This will also be used for the .lib, .lef, .v, and .spef files

# that are generated by this script

set BASENAME "CONTADOR_cont1"


# set the name of the filler and clock buffer cells - you don't
need a list

# if you only have one

set fillerCells [list FILL1 FILL2 FILL5 FILL10 FILL25]

set clockBufName [list CLKBU12 CLKBU15 CLKBU2 CLKBU4 CLKBU6
CLKBU8 CLKIN0 CLKIN1 CLKIN10 CLKIN12 CLKIN15 CLKIN2 CLKIN3
CLKIN4 CLKIN6 CLKIN8] ;# Footprint of inverter in .lib file


# choose numbers that make sense for you

set usepct 0.70 ;# percent utilization in placing cells

set rowgap 30   ;# gap between pairs of std cell rows

set aspect 0.50 ;# aspect ratio of overall cell (1.0 is square)

# less than 1.0 is landscape, greater than 1.0 is portrait


# Set some values that define the power rings and stripes.

# use these defaults, or choose your own.

set pwidth 8    ;# power rail width

set pspace 2    ;# power rail space

set swidth 8    ;# power stripe width

set sspace 2    ;# power stripe spacing

set sspacing 200    ;# power stripe spacing between stripes

set soffset 200   ;# power stripe offset to first stripe

set coregap 22  ;# gap between the core and the power rails


############################################################

# SCRIPT CONFIGURATION

############################################################

set init_des 1 ;#Set if you wish to initialize the design

set fplan 1    ;#Set if you wish to create the floorplan (might
be done by hand...)

set pplan 1    ;#Set if you wish to create the power rings and
stripes

set place 1    ;#Set if you wish to place the cells and optimize
(pre-CTS)

set cts 1      ;#Set if you wish to create the clock tree, and
optimize (post-CTS)

set route 1    ;#Set if you wish to route the design using
nanoRoute

set verify 1   ;#Set if you wish to verify the design and
produce output files

set exit_end 0 ;#Set if you wish to exit the program when the
script is ended
```

Bibliografía

- Notas prácticas de la asignatura.
- Manuales de las herramientas.
- *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*. Erik Brunvand. Addison-Wesley. 2010.
- *Application-Specific Integrated Circuits.* Michael John Sebastian Smith. Addison-Wesley. 1997.



Slides Credits

- Digital IC-Proyect and Verification. ASIC Synthesis. Deepak Dasalukunte & Joachim Rodriges. Lund University.
- Design Compiler Tutorial. http://www.tkt.cs.tut.fi/tools/public/tutorials/synopsys/design_compiler/gsdc.html Tampere University of Technology.
- *Application-Specific Integrated Circuits.* Michael John Sebastian Smith. Addison-Wesley. 1997.